Package: walk (via r-universe)

November 5, 2024

Title Model Animal Movement With Continuous-Time Markov Chains

Version 0.0.0.920

Date 2024-04-15

Description Model animal movement using continuous-time Markov chain models.

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports TMB (>= 1.9.0), Matrix, terra, dplyr, mvtnorm, numDeriv, optimx, sf, spdep, ctmm, methods, tidyr, rARPACK

LinkingTo TMB, Rcpp, RcppArmadillo, expQ2

Remotes dsjohnson/expQ2

NeedsCompilation yes

Depends R (>= 2.10)

LazyData true

Config/pak/sysreqs cmake libgdal-dev gdal-bin libgeos-dev libgmp3-dev libgsl0-dev libicu-dev libmpfr-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://dsjohnson.r-universe.dev

RemoteUrl https://github.com/dsjohnson/walk

RemoteRef HEAD

RemoteSha 5745638f84b20ee6abe25e8fd060157547885216

Contents

arg_funs 2
ctmc_n2ll 3
dm_matrix 4
fit_ctmc 4

gen_invLogit 5
get_betas 6
get_lim_ud 6
get_Q 7
make_walk_data 7
predict_ctmc 8
proc_telem 9
set_barrier_mask 10
softPlus 10

Index **11**

`arg_funs` *Define specifics for model structure and fitting control*

Description

Define specifics for model structure and fitting control

Usage

```
ctmc_model(form = ~1, link = "soft_plus", a = 1, L = 0, U = 0)

ctmc_control(
  q_r = ctmc_model(),
  q_m = ctmc_model(),
  p = FALSE,
  delta = "uniform",
  form = "mult",
  norm = TRUE
)
```

Arguments

`form`	The form of the rate matrix entries. Can be one of: `"mult"` for residency times movement, i.e., q_ij = q_r(i) * pi_m(i,j), `"add"` for a model of the form q_ij = q_r(i) + pi_m(i,j), and `"sde"` for a Langevin diffusion approximation. The `"sde"` is very similar to the additive model.
`link`	A link funtion for the residency or movement model. One of `"soft_plus"` or `"log"`.
`a`	Scale parameter for the `"soft_plus"` link function. Ignored for `link="log"`. The a parameter determines the approximation to a hard plus function, i.e., as a becomes large the soft plus function converges to `g^{-1}(x)` = `max(0,x)`. For this specification, a must be greater than or equal to 1.
`q_r`	A named list for the residency model. Must contain elements: `form`, `link`, and a. The easiest way to construct this is the `ctmc_model` function.
`q_m`	A named list for the movement portion of the model. See `q_r`.

`p`	Logical. Should a zero-inflation parameter be used. Necessary when there might be outliers when the location likelihood does not overlap likely locations.
`delta`	Vector of probabilities for the initial location. Can be one of `"uniform"` for a uniform distribution over all cells, `"stationary"` to use the stationary distribution implied by the parameters, or a numeric vector the same length as the number of cells.
`norm`	Should the movement portion be adjusted to sum to 1. This is the parameterization suggested by Hewitt et al. (2023).

References

Hewitt, J., Gelfand, A. E., & Schick, R. S. (2023). Time-discretization approximation enriches continuous-time discrete-space models for animal movement. The Annals of Applied Statistics, 17:740-760.

`ctmc_n2ll` *Evaluate movement MMPP log-likelihood*

Description

Evaluate movement MMPP log-likelihood

Usage

```
ctmc_n2ll(par, data_list, check_rho = NULL, debug = 0, ...)
```

Arguments

`par`	Parameter vector
`data_list`	List of required data objects to evaluate likelihood
`check_rho`	Check if rho is too big for uniformitazation calculation of expQ. Value is the size of rho to check.
`debug`	For developers only, leave in the default setting.
`...`	Extra wiggle room for ignored arguments.

Author(s)

Devin S. Johnson

`dm_matrix` *Create sparse representations of design matrices for movement and resight models*

Description

Create sparse representations of design matrices for movement and resight models

Usage

```
dm_q_m(formula, ddl)

dm_q_r(formula, ddl)
```

Arguments

`formula`	An `R` formula object for the paramter vector.
`ddl`	Design data list

Details

This function is not designed for end-users but is exported for posterity.

`fit_ctmc` *Fit CTMC movement model to telemetry data*

Description

Fit CTMC movement model to telemetry data

Usage

```
fit_ctmc(
  walk_data,
  model_parameters = ctmc_control(),
  pen_fun = NULL,
  hessian = TRUE,
  reals = FALSE,
  start = NULL,
  method = "nlminb",
  fit = TRUE,
  eq_prec = 1e-08,
  check_rho = NULL,
  debug = 0,
  ...
)
```

Arguments

`walk_data`	A design data list produced by the function `make_walk_data`.
`model_parameters`	A named list giving specifications for the model. See `ctmc_control`.
`pen_fun`	An optional penalty function. Should be on the scale of a log-prior distribution.
`hessian`	Logical. Should the Hessian matrix be calculated to obtain the parameter variance-covariance matrix.
`reals`	Calculate real values for expected residency, cell transition probabilities, and outlier proportion for observations.
`start`	Optional starting values for the parameter must be a list of the form `list(beta_l=c(), beta_q_r=c(), beta_q_r=c())`.
`method`	Optimization method. See `optimr`
`fit`	Logical. Should the likelihood be optimized?
`eq_prec`	Error rate of matrix exponential calculation. Defaults to `1.0e-8`. This is a generous value. If the model is running slow, you can try reducing it to, say, `1.0e-4`.
`check_rho`	Check if rho is too big for uniformitazation calculation of expQt. Value is the size for which rho is too big, causing likelihood calculation error.
`debug`	Integer from 1-4. Opens browser() at various points in the function call. Mostly for package developers.
`...`	Additional arguments passed to the optimization function `optimr` from the `optimx-package`.

Author(s)

Devin S. Johnson

`gen_invLogit` *General inverse logit function*

Description

General inverse logit function

Usage

```
gen_invLogit(x, L = 0, U = 1)
```

Arguments

`x`	numeric value
`L`	Lower bound
`U`	Upper bound

`get_betas` *Summarize beta parameter values*

Description

Takes fitted model and produces a summary of the beta estimates.

Usage

```
get_betas(par, V = NULL, data_list)
```

Arguments

`par`	Model estimated parameters
`V`	Optional variance-covariance matrix for the parameters. If none is given no standard errors will be provided for the real parameter estimates
`data_list`	The data list from a fitted model object

Author(s)

Devin S. Johnson

`get_lim_ud` *Get the limiting utilization distribution of the CTMC movement process*

Description

Get the limiting utilization distribution of the CTMC movement process

Usage

```
get_lim_ud(fit = NULL, hpd = NULL, method = "lu", ...)
```

Arguments

`fit`	A moveMMPP fitted model object from `fit_mmpp_dir`.
`hpd`	A vector of probabilities. Will return columns with highest probability area for each specified probability. E.g., `hpd=c(0.5, 0.95)` will return 2 extra columns with 50 and 95% HPD densities.
`method`	Method used for eigen decomposition. One of `"lu"` or `"arpack"`.
`...`	Extra arguments to pass to `eigs`

Author(s)

Devin S. Johnson

`get_Q` *Get movement transition matrix from fitted moveMMP object*

Description

Get movement transition matrix from fitted moveMMP object

Usage

```
get_Q(fit, sparse = TRUE)
```

Arguments

`fit`	A moveMMPP fitted model object from `fit_mmpp_dir`.
`sparse`	Logical. Should the matrix be returned in a sparse format from the `Matrix` package. Defaults to `sparse` = `TRUE`.

Author(s)

Devin S. Johnson

`make_walk_data` *Produce design data for use in fitting MMPP movement models*

Description

Produce design data for use in fitting MMPP movement models

Usage

```
make_walk_data(
  proc_data,
  cell_data,
  grad = NULL,
  rast_mask = NULL,
  directions = "rook",
  debug = 0,
  ...
)
```

Arguments

`proc_data`	A sparse matrix with rows corresponding to time steps and columns corresponding to cells. The entries are probabilities that the animal is located in the corresponding cell and time.
`cell_data`	A `SpatRaster` object from the terra package.
`grad`	A character vector of names of `cell_data` layers for which gradient covariates will be constructed.
`rast_mask`	Raster mask for inaccessible cells when `cell_data` is of type `SpatRaster` from the `terra` package. This is ignored if `cell_data` is an `POLYGON` data frame from the `sf` package.
`directions`	—.
`debug`	Debugging level: 1-3 mainly for package developers.
`...`	Ignored arguments.

`predict_ctmc` *Predict Movement Path From Fitted CTMC Model*

Description

Predict Movement Path From Fitted CTMC Model

Usage

```
predict_ctmc(
  fit,
  walk_data,
  aux_timestamp = NULL,
  trunc_tol = 1e-08,
  debug = 0,
  ...
)
```

Arguments

`fit`	A fitted model object produced by `fit_ctmc`
`walk_data`	A design data list produced by the function `make_walk_data`.
`aux_timestamp`	Additional POSIX times for which location prediction is desired.
`trunc_tol`	Level for truncating small location prediction probabilities.
`debug`	Developer debugging.
`...`	Aditional arguments passed to internal function `ctmc_predict_arma`.

Author(s)

Devin S. Johnson

References

Hewitt, J., Gelfand, A. E., & Schick, R. S. (2023). Time-discretization approximation enriches continuous-time discrete-space models for animal movement. The Annals of Applied Statistics, 17:740-760.

`proc_telem` *Convert ctmm telemetry data object to a terra* `SpatRaster` *stack*

Description

Convert ctmm telemetry data object to a terra `SpatRaster` stack

Usage

```
proc_telem(
  data,
  cell_data,
  time_unit = "hours",
  return_type = "sparse",
  max_err = NULL,
  trunc = 1e-08
)
```

Arguments

`data`	A `telemetry` object from the ctmm package
`cell_data`	A `\link[terra]{SpatRaster}` stack of covariates that will be used in CTMC movement modeling. Cells with `NA` values will be considered areas to the animal cannot travel, i.e., likelihood surfaces will be `0` for those cells.
`time_unit`	Unit of measurement for numeric conversion of the POSIX timestamps in the telemetry data. Defaults to `"hours"`. Can specify, e.g., `"seconds"` or `"days"`
`return_type`	Type of object returned. One if `"data.frame"`, `"sparse"` (sparse matrix), `"matrix_df"` (matrix form of `"data.frame"`), or `"dense"` (dense matrix).
`max_err`	The maximum error in meters. If unspecified it will be set to 4 times the maximum error standard deviation as determined by the UERE and HDOP of the telemetry data.
`trunc`	The smallest probability value that is considered to be > 0. Defaults to 1.0e-8.

Details

This function takes the HDOP information in the `telemetry` object to produce a `SpatRaster` likelihood surface over the `SpatRaster` defined by the `raster` argument for each location. This can then be passed to CTMC HMM fitting functions.

Author(s)

Devin S. Johnson

`set_barrier_mask` *Denote cells where the animal cannot travel*

Description

Denote cells where the animal cannot travel

Usage

```
set_barrier_mask(x, y, overlap = 0.95)
```

Arguments

`x`	a polygon object that denotes places where the animal cannot travel
`y`	A `SpatRaster` where cells will be set to `NA` that are inside the polygon object.
`overlap`	The amount of overlap betwen the raster cell and polygon such that the cell is considered "inside" the polygon. Defaults to `overlap = 0.95`. If `overlap = 1` then a cell has to be completely inside the polygon to be maked.

`softPlus` *Soft-plus function*

Description

Soft-plus function

Usage

```
softPlus(x, a = 1)
```

Arguments

`x`	numeric value
`a`	scale parameter, a must be >1.

Index

arg_funs, 2

ctmc_control, *5*
ctmc_control *(*arg_funs*)*, 2
ctmc_model *(*arg_funs*)*, 2
ctmc_n2ll, 3

dm_matrix, 4
dm_q_m *(*dm_matrix*)*, 4
dm_q_r *(*dm_matrix*)*, 4

eigs, *6*

fit_ctmc, 4, *8*
fit_mmpp_dir, *6*, *7*

gen_invLogit, 5
get_betas, 6
get_lim_ud, 6
get_Q, 7

make_walk_data, *5*, 7, *8*

optimr, *5*

predict_ctmc, 8
proc_telem, 9

set_barrier_mask, 10
softPlus, 10